

Machine Learning Review

Akshat Pandey

Contents

1	Logistic Regression	2
1.1	Derivations	2
2	Softmax Regression (Multinomial Logistic Regression)	4
2.1	Derivations	4
3	Linear Regression	7
3.1	Derivations	7
4	Decision Tree Classifier/Regression	8
4.1	Regression	8
4.2	Classification	8
5	Ensemble Learning	9
5.1	Random Forests	9
5.2	Gradient Boosted Trees	9
5.2.1	Derivations	9

1 Logistic Regression

$$h(z) = \frac{1}{1 + e^{-z}} \quad [\text{Logistic Function}]$$

$$L_{\text{CE}}(y, h(z)) = -[y \log h(z) + (1 - y) \log(1 - h(z))] \quad [\text{Cross Entropy Loss}]$$

$$w_i = w_i - \alpha \times \frac{\partial}{\partial w_i} L_{\text{CE}}(y_i, h(z_i)) \quad [\text{Weight Update}]$$

$$= w_i - \alpha \times (x_i(h(z_i) - y_i))$$

$$w = w - \alpha \times \frac{1}{B} \sum_{i=1}^B x_i(h(z_i) - y_i) \quad [\text{Batch Weight Update}]$$

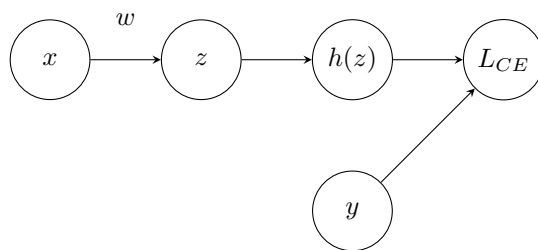


Figure 1: Logistic Regression Computation Graph

1.1 Derivations

Derivative of Logistic Function

$$\begin{aligned} \frac{\partial}{\partial z} h(z) &= \frac{\partial}{\partial z} \frac{1}{1 + e^{-z}} \\ &= \frac{\partial}{\partial z} (1 + e^{-z})^{-1} \\ &= -(1 + e^{-z})^{-2} \left(\frac{\partial}{\partial z} 1 + e^{-z} \right) \\ &= -(1 + e^{-z})^{-2} \left(\frac{\partial}{\partial z} e^{-z} \right) \\ &= -(1 + e^{-z})^{-2} \left(e^{-z} \frac{\partial}{\partial z} - z \right) \\ &= -(1 + e^{-z})^{-2} (-e^{-z}) \\ &= \frac{e^{-z}}{(1 + e^{-z})^2} \end{aligned}$$

$$\begin{aligned}
&= \frac{e^{-z}}{(1 + e^{-z})} \frac{1}{(1 + e^{-z})} \\
&= \left(\frac{1 + e^{-z}}{(1 + e^{-z})} - \frac{1}{(1 + e^{-z})} \right) \frac{1}{(1 + e^{-z})} \\
&= \left(1 - \frac{1}{(1 + e^{-z})} \right) \frac{1}{(1 + e^{-z})} \\
&= (1 - h(z))h(z)
\end{aligned}$$

Derivative of Logistic Function with respect to weight w_i

$$\begin{aligned}
z_i &= w_i \times x_i \\
\frac{\partial}{\partial w_i} z_i &= \frac{\partial}{\partial w_i} w_i \times x_i \\
&= x_i \\
\frac{\partial}{\partial w_i} h(z_i) &= \frac{\partial z_i}{\partial w_i} \frac{\partial h(z_i)}{\partial z_i} \\
&= x_i (1 - h(z_i)) h(z_i)
\end{aligned}$$

Derivative of Cross Entropy Loss with respect to weight w_i

$$\begin{aligned}
\frac{\partial}{\partial w_i} L_{\text{CE}}(y_i, h(z_i)) &= \frac{\partial}{\partial w_i} - [y_i \log h(z_i) + (1 - y_i) \log(1 - h(z_i))] \\
&= - \left[y_i \frac{\partial}{\partial w_i} \log h(z_i) + (1 - y_i) \frac{\partial}{\partial w_i} \log(1 - h(z_i)) \right] \\
&= - \left[\frac{y_i}{h(z_i)} \frac{\partial}{\partial w_i} h(z_i) + \frac{(1 - y_i)}{(1 - h(z_i))} \frac{\partial}{\partial w_i} (1 - h(z_i)) \right] \\
&= - \left[\frac{y_i}{h(z_i)} \frac{\partial}{\partial w_i} h(z_i) - \frac{(1 - y_i)}{(1 - h(z_i))} \frac{\partial}{\partial w_i} h(z_i) \right] \\
&= - \left[\frac{\partial}{\partial w_i} h(z_i) \left(\frac{y_i}{h(z_i)} - \frac{(1 - y_i)}{(1 - h(z_i))} \right) \right] \\
&= - \left[x_i (1 - h(z_i)) h(z_i) \left(\frac{y_i}{h(z_i)} - \frac{(1 - y_i)}{(1 - h(z_i))} \right) \right] \\
&= - [x_i (y_i (1 - h(z_i)) - (1 - y_i) h(z_i))] \\
&= - [x_i (y_i - h(z_i))] \\
&= x_i (h(z_i) - y_i)
\end{aligned}$$

2 Softmax Regression (Multinomial Logistic Regression)

$$S(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad [\text{Softmax Function}]$$

$$L_{\text{CE}}(y, S(z)) = - \sum_{j=1}^K y_j \log S(z_j) \quad [\text{Cross Entropy Loss}]$$

$$\begin{aligned} w_i &= w_i - \alpha \times \frac{\partial}{\partial w_i} L_{\text{CE}}(y_i, S(z_i)) \quad [\text{Weight Update}] \\ &= w_i - \alpha \times x_i (S(z_i) - y_i) \end{aligned}$$

$$w = w - \alpha \times \frac{1}{B} \sum_{i=1}^B x_i (S(z_i) - y_i) \quad [\text{Batch Weight Update}]$$

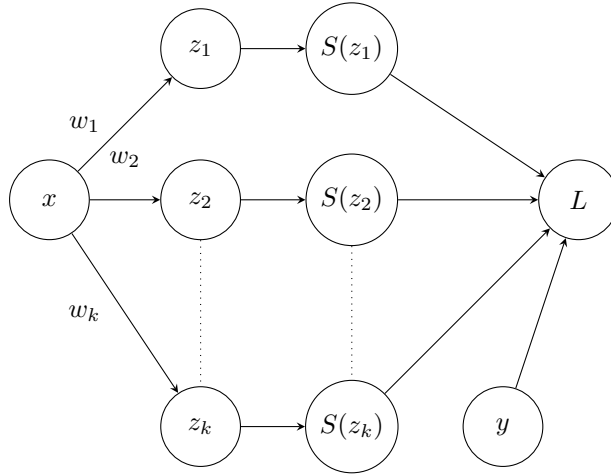


Figure 2: Softmax Regression Computation Graph

2.1 Derivations

Derivative of arbitrary sigmoid output $S(z_j)$ with respect to arbitrary linear combination output z_i :

$$\frac{\partial S(z_j)}{\partial z_i} = \frac{\partial}{\partial z_i} \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$$

$$\begin{aligned}
&= \frac{\frac{\partial}{\partial z_i} e^{z_j} (\sum_{k=1}^K e^{z_k}) - e^{z_j} (\frac{\partial}{\partial z_i} \sum_{k=1}^K e^{z_k})}{(\sum_{k=1}^K e^{z_k})^2} \\
&= \begin{cases} \frac{\frac{\partial}{\partial z_i} e^{z_j} (\sum_{k=1}^K e^{z_k}) - e^{z_j} (\frac{\partial}{\partial z_i} \sum_{k=1}^K e^{z_k})}{(\sum_{k=1}^K e^{z_k})^2} & i = j \\ \frac{\frac{\partial}{\partial z_i} e^{z_j} (\sum_{k=1}^K e^{z_k}) - e^{z_j} (\frac{\partial}{\partial z_i} \sum_{k=1}^K e^{z_k})}{(\sum_{k=1}^K e^{z_k})^2} & i \neq j \end{cases} \\
&= \begin{cases} \frac{e^{z_i} (\sum_{k=1}^K e^{z_k}) - (e^{z_i})^2}{(\sum_{k=1}^K e^{z_k})^2} & i = j \\ \frac{0 - e^{z_j} e^{z_i}}{(\sum_{k=1}^K e^{z_k})^2} & i \neq j \end{cases} \\
&= \begin{cases} S(z_i)(1 - S(z_i)) & i = j \\ -S(z_j)S(z_i) & i \neq j \end{cases} \\
&= S(z_i)(\delta_{i,j} - S(z_j))
\end{aligned}$$

Derivative of loss with respect to arbitrary linear combination output z_i :

$$\begin{aligned}
\frac{\partial L}{\partial z_i} &= - \sum_{k=1}^K y_k \log S(z_k) \\
&= - \left[\frac{y_i}{S(z_i)} S(z_i)(1 - S(z_i)) - \sum_{k \neq i}^K \frac{y_k}{S(z_k)} (S(z_k)S(z_i)) \right] \\
&= - \left[y_i(1 - S(z_i)) - \sum_{k \neq i}^K y_k S(z_i) \right] \\
&= - \left[y_i - S(z_i)y_i - \sum_{k \neq i}^K S(z_i)y_k \right] \\
&= - \left[y_i - \sum_{k=1}^K S(z_i)y_k \right]
\end{aligned}$$

$$\begin{aligned}
&= - \left[y_i - S(z_i) \sum_{k=1}^K y_k \right] \\
&= S(z_i) - y_i
\end{aligned}$$

Derivative of weight w_i with respect to linear combination z_i :

$$\begin{aligned}
\frac{\partial z_i}{\partial w_i} &= \frac{\partial}{\partial w_i} w_i \times x_i \\
&= x_i
\end{aligned}$$

Derivative of weight w_i with respect to loss:

$$\begin{aligned}
\frac{\partial L}{\partial w_i} &= \frac{\partial z_i}{\partial w_i} \frac{\partial L}{\partial z_i} \\
&= x_i (S(z_i) - y_i)
\end{aligned}$$

3 Linear Regression

$$\begin{aligned}
 h(x) &= w^T \cdot x \\
 L_{\text{MSE}}(y, h(x)) &= (y - h(x))^2 && [\text{Mean Squared Error Loss}] \\
 w_i &= w_i - \alpha \times \frac{\partial}{\partial w_i} L_{\text{MSE}}(y_i, h(x_i)) && [\text{Weight Update}] \\
 &= w_i - \alpha \times x_i(h(x_i) - y_i) \\
 w &= w - \alpha \times \sum_{i=0}^B x_i(h(x_i) - y_i)
 \end{aligned}$$

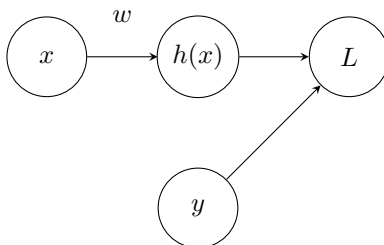


Figure 3: Linear Regression Computation Graph

3.1 Derivations

Derivative of Mean Squared Error with respect to weight w_i

$$\begin{aligned}
 \frac{\partial}{\partial w_i} L_{\text{MSE}}(y_i, h(x_i)) &= \frac{\partial}{\partial w_i} (y_i - h(x_i))^2 \\
 &= 2(y_i - h(x_i)) \times \frac{\partial}{\partial w_i} (y_i - h(x_i)) \\
 &= 2(y_i - h(x_i)) \times \left(\frac{\partial}{\partial w_i} y_i - \frac{\partial}{\partial w_i} h(x_i) \right) \\
 &= 2(y_i - h(x_i)) \times \left(-\frac{\partial}{\partial w_i} w_i \times x_i \right) \\
 &= -2x_i(y_i - h(x_i)) \\
 &= 2x_i(h(x_i) - y_i) \\
 &\propto x_i(h(x_i) - y_i)
 \end{aligned}$$

4 Decision Tree Classifier/Regression

$$\begin{aligned}
 N_m &= \#\{x_i \in R_m\} && \text{[Number of points in node]} \\
 j &= \text{attribute} \\
 s &= \text{value}
 \end{aligned}$$

4.1 Regression

$$\hat{c}_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i \quad \text{[Output]}$$

$$j, s = \min_{j,s} \left[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right] \quad \text{[Splitting Condition]}$$

4.2 Classification

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k) \quad \text{[Probability of class } k \text{ in node } m]$$

$$\hat{c}_m = \max_k \hat{p}_{mk} \quad \text{[Output]}$$

$$j, s = \min_{j,s} \left[R_1(j, s) \sum_{k=1}^K \hat{p}_{1k} (1 - \hat{p}_{1k}) + R_2(j, s) \sum_{k=1}^K \hat{p}_{2k} (1 - \hat{p}_{2k}) \right] \quad \text{[Gini Index Splitting condition]}$$

$$j, s = \min_{j,s} \left[-R_1(j, s) \sum_{k=1}^K \hat{p}_{1k} \log \hat{p}_{1k} - R_2(j, s) \sum_{k=1}^K \hat{p}_{2k} \log \hat{p}_{2k} \right] \quad \text{[Cross Entropy Splitting condition]}$$

5 Ensemble Learning

5.1 Random Forests

```
1 for b = 1 to B:  
2     Retrieve a bootstrap sample  
3     Grow a full decision tree on the sample with random features  
4 return ensemble
```

Figure 4: Random Forest construction process

$$f(x) = \frac{1}{B} \sum_{b=1}^B T_b(x) \quad [\text{Regression}]$$

$$f(x) = \text{majority vote}\{T_b(x)\}_1^B \quad [\text{Classification}]$$

5.2 Gradient Boosted Trees

```
1 fit tree to data  
2 calculate gradient  
3  
4 while loss not acceptable:  
5     fit new tree to negative gradient
```

$$s(x_i) = \frac{e^{x_i}}{\sum_{k=1}^K e^{x_k}} \quad [\text{Softmax Function}]$$

$$L_{\text{CE}}(y, \hat{y}) = - \sum_{k=1}^K y \log \hat{y} \quad [\text{Cross Entropy Loss}]$$

$$\begin{aligned} f(x) &= f(x) - \alpha \frac{\partial L(y, f(x))}{\partial f(x)} \\ &= f(x) - \alpha [s(f(x)) - y] \end{aligned} \quad [\text{Model update}]$$

5.2.1 Derivations

Derivative of softmax output p_j with respect to single $f(x)$ output o_i to:

$$\frac{\partial p_j}{\partial o_j} = \frac{\partial}{\partial o_j} \frac{e^{o_j}}{\sum_{k=1}^K e^{o_k}}$$

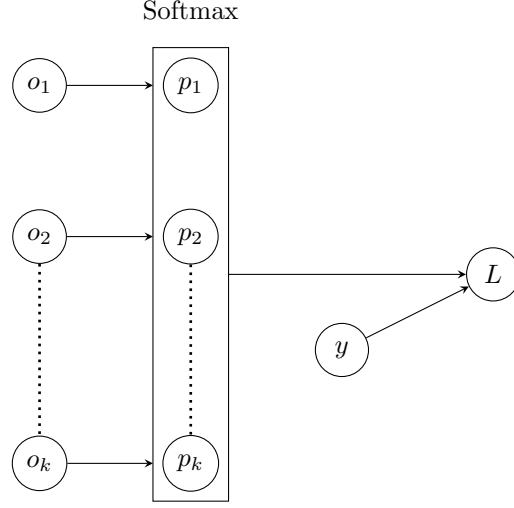


Figure 5: Softmax computation graph for gradient boosted tree output

$$\begin{aligned}
&= \frac{(\frac{\partial}{\partial o_i} e^{o_j})(\sum_{k=1}^K e^{o_k}) - (e^{o_j})(\frac{\partial}{\partial o_i} \sum_{k=1}^K e^{o_k})}{(\sum_{k=1}^K e^{o_k})^2} \\
&= \begin{cases} i = j & \frac{e^{o_j} \sum_{k=1}^K e^{o_k} - (e^{o_j})^2}{(\sum_{k=1}^K e^{o_k})^2} \\ i \neq j & \frac{0 - e^{o_j} e^{o_i}}{(\sum_{k=1}^K e^{o_k})^2} \end{cases} \\
&= \begin{cases} i = j & p_j - p_j^2 \\ i \neq j & -p_j p_i \end{cases}
\end{aligned}$$

Derivative of loss L with with respect to single $f(x)$ output o_i to:

$$\begin{aligned}
\frac{\partial L}{\partial o_i} &= \frac{\partial}{\partial o_i} \left[- \sum_{k=1}^K y_k \log p_k \right] \\
&= - \left[\sum_{k=1}^K \frac{\partial}{\partial o_i} y_k \log p_k \right] \\
&= - \left[\sum_{k=1}^K \frac{y_k}{p_k} \frac{\partial}{\partial o_i} p_k \right] \\
&= - \left[\frac{y_i(p_i - p_i^2)}{p_i} + \sum_{k \neq i}^K \frac{-y_k(p_k p_i)}{p_k} \right]
\end{aligned}$$

$$\begin{aligned}
&= - \left[y_i(1 - p_i) - \sum_{k \neq i}^K y_k p_i \right] \\
&= - \left[y_i - p_i y_i - p_i \sum_{k \neq i}^K y_k \right] \\
&= - \left[y_i - p_i \sum_{k=1}^K y_k \right] \\
&= - [y_i - p_i] \\
&= p_i - y_i
\end{aligned}$$